REMOVING ARCS FROM A NETWORK

by

Richard Wollmer

OPERATIONS RESEARCH CENTER

INSTITUTE OF ENGINEERING RESEARCH

UNIVERSITY OF CALIFORNIA - BERKELEY

REMOVING ARCS FROM A NETWORK

by

Richard Wollmer
Operations Research Center
University of California, Berkeley

March 13, 1964

ORC 64-5(RR)

This research was primarily supported by the Western Management Science Institute of the University of California, Los Angeles.

REMOVING ARCS FROM A NETWORK

I. Introduction

This paper is concerned with a sensitivity analysis on a maximum flow network.

The network is defined by a set of arcs and a set of points called nodes. Each arc joins two nodes and has associated with it a positive capacity which represents the maximum amount of flow that may pass over it. One of the nodes is designated as the source and another as the sink.

From these nodes, arcs, and capacities, the maximum amount of flow that may pass from source to sink may be calculated.(1) The question arises as to what happens to the network if a number of arcs are to be removed. Specifically, given a maximum flow network from which n arcs are to be removed, which n arcs, if removed, would reduce the maximum flow from source to sink the most and what would be the resulting maximum flow? This paper presents an algorithm for solving such a problem for a certain class of networks.

The algorithm would be helpful in determining how sensitive a transportation system might be to having some of its roads closed down for repairs or tied up by traffic accidents. It might also shed light on the problem of adding arcs which is of interest in deciding where to build new roads.

---

1) pp. 17-22

II. Max-Flow Min-Cut Theorem

The central idea in the solving of maximum flow network problems is summarized in the max-flow min-cut theorem. First, it is necessary to define a cut set and its value.

Definition: Consider a network consisting of nodes which include a source and a sink, and capacitated arcs which join two nodes. Let A and B be a partition of the nodes such that the source is in A and the sink is in B . Then the set of arcs which join a node in A to a node in B is called a cut set and is denoted [A, B]. Furthermore, the value of this cut set, V[A, B], is equal to the sum of the capacities of its arcs.

A property of any cut set, [A, B], is that any path from source to sink must use at least one of its arcs. This is true since the source is in A , the sink in B , and hence any path connecting the two must contain an arc joining a node of A to a node of B . Thus, it is apparent that the maximum flow cannot exceed the minimum value of all cut sets. The max-flow min-cut theorem states that the maximum flow actually equals the minimum cut.

III. The Topological Dual

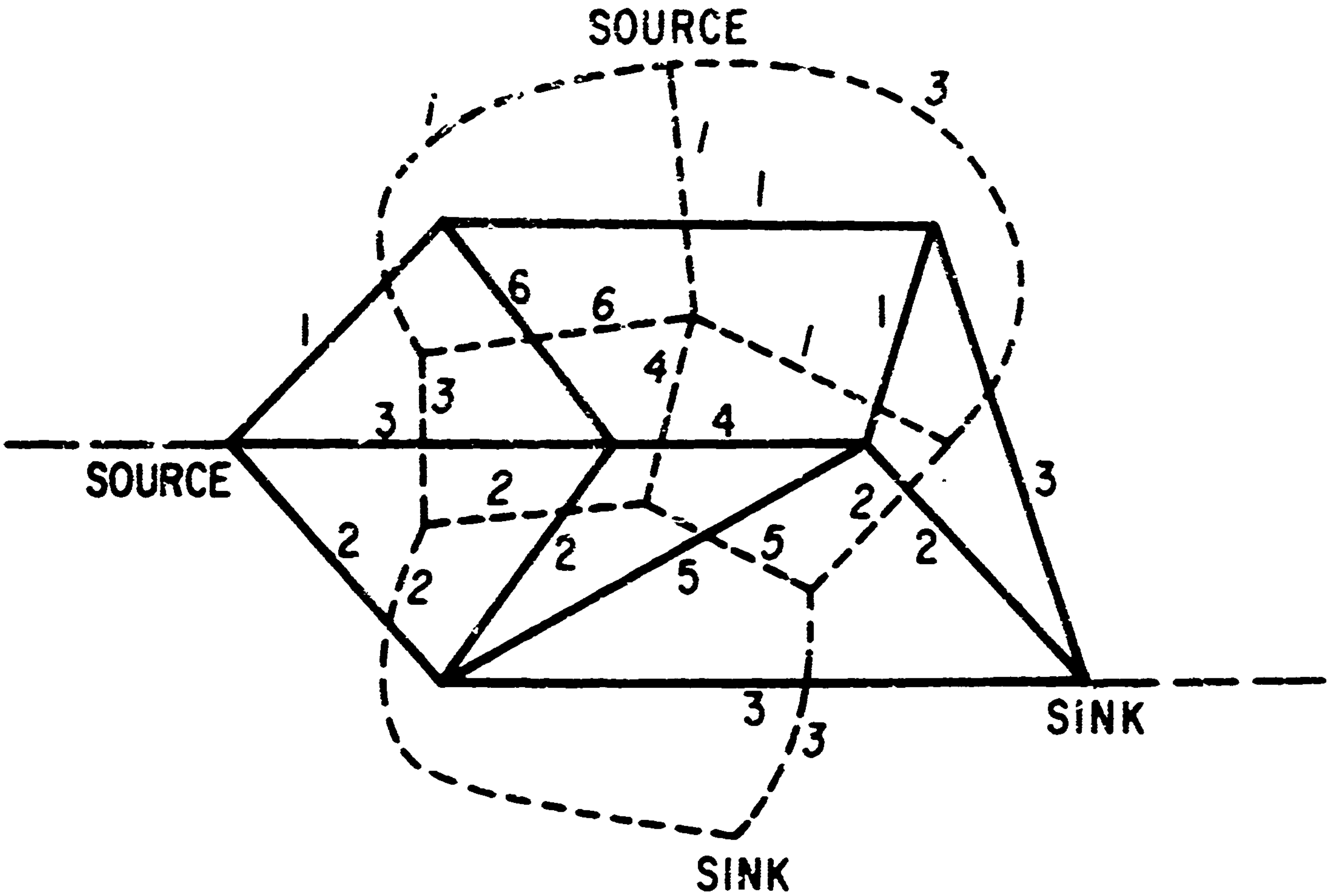
The topological dual of a network, when defined, is another network in which the arcs, instead of having capacities, have lengths. Furthermore, there is a one-to-one correspondence between the cuts of the original network and the routes through the dual, and the problem of finding the minimum cut may be reduced to one of finding a shortest route.

Let the original maximum flow network be called the primal. To the primal add two artificial arcs, one extending from the source to minus infinity and the other from the sink to plus infinity. The resulting network will be referred to as the modified primal. The dual is defined if and only if the modified primal is planar, a planar network being one that can be drawn on a plane such that no two arcs intersect except at a node.

When defined, the dual is constructed in the following manner:

1. Place a node in each mesh of the modified primal. Let the source be the node in the mesh above the modified primal and the sink be the node in the mesh below it.
2. For each arc in the primal construct an arc that intersects it and joins the nodes in the meshes on either side of it.
3. Assign each arc of the dual a length equal to the capacity of the primal arc it intersects.

An example of a network and its dual is shown in Figure 1. Figure 2 shows an example of a network where the modified primal is not planar and hence the dual is not defined. Let a route through the dual be any path from its source to its sink. It follows that there is a one-to-one correspondence between the cuts of the primal and the routes of the dual.



a arc of primal with capacity a

artificial arc

b arc of dual with length b

FIG. I A NETWORK AND ITS DUAL

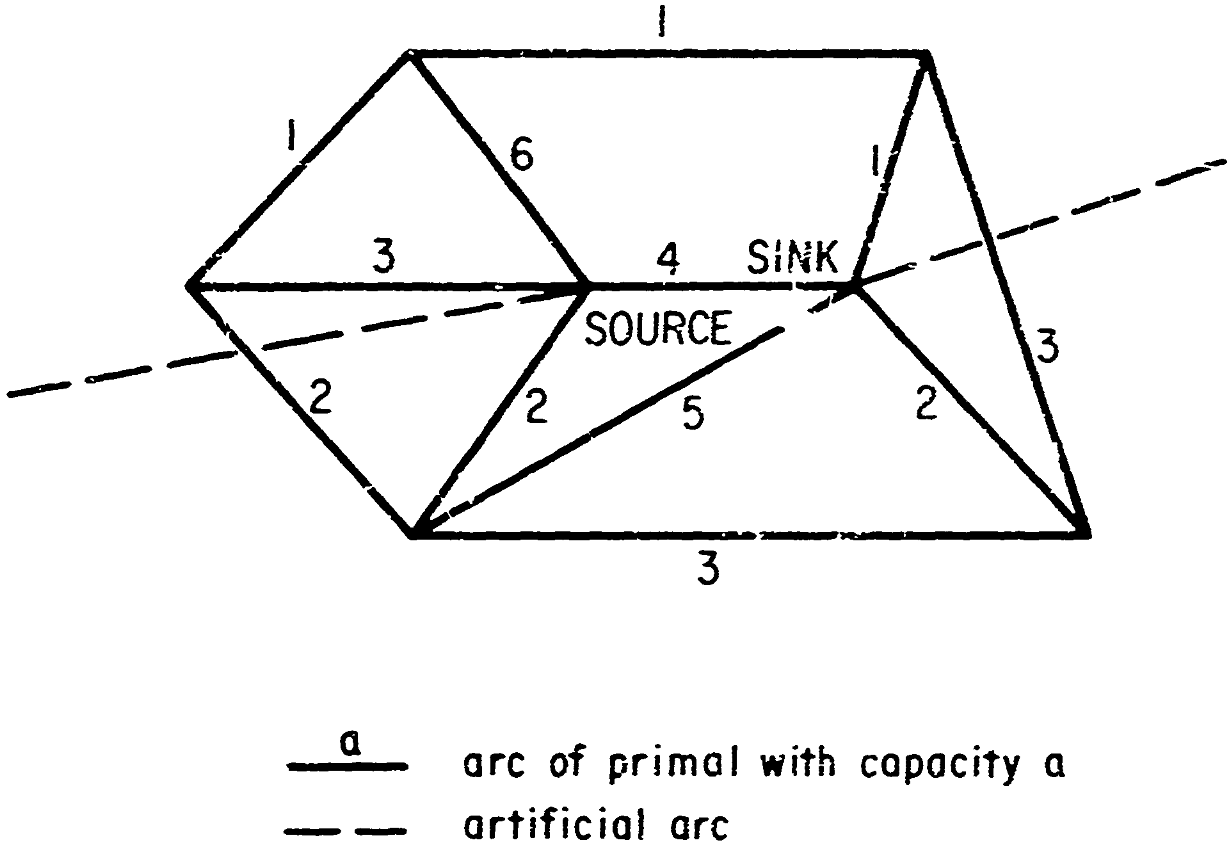


FIG. 2 A NETWORK WHERE THE MODIFIED PRIMAL IS NOT PLANAR AND HENCE THE DUAL IS NOT DEFINED. NOTE, HOWEVER, THAT THE PRIMAL IS PLANAR.

THEOREM: Given a network such that the modified primal is planar, consider any route through the dual. The arcs of the primal which intersect this route form a cut whose value is equal to the length of the route. Conversely, for each cut of the primal there is a route through the dual whose length is equal to the value of that cut, and such that the cut forms the set of arcs in the primal that intersect this route.

PROOF: Consider any route through the dual. Let A be the set of all nodes that lie to the left of this route and B the set of all nodes to the right of it. From the construction of the dual it follows that the source is in A and the sink in B. It follows that any arc connecting a node of A to a node of B is connecting nodes which lie on opposite sides of this route and must therefore intersect it. Furthermore, any arc which intersects this route connects nodes on opposite sides of it and must therefore be connecting a node of A to a node of B. Hence the cut set [A, B] is the set of arcs which intersect this route. Since the capacity of any arc of [A, B] is equal to the length of the arc in the dual it intersects and conversely, it follows that the value of the cut [A, B] is equal to the total length of this route. Now consider any cut set [A, B]. The set of arcs in the dual which intersect it forms a route. Also, all the nodes in A lie to the left of this route and all the nodes of B to the right of it. Hence the value of cut [A, B] is equal to the length of this route.

It follows from the above theorem that finding the value

of the minimum cut of the primal is equivalent to finding the length of the minimum route through the dual.

IV. The Algorithm

Since the maximum flow through a network is equal to the minimum cut which in turn is equal to the shortest route through the dual, it follows that the problem of finding the n arcs which when removed from a network will reduce its capacity the most is equivalent to finding those n arcs of the dual which when assigned zero length will make the minimum route as short as possible.

It is the latter problem that the algorithm of this section solves. Throughout the rest of this report all nodes and arcs refer to the dual network unless specified otherwise.

Let an i-arc path to node a be any path from the source to node a with the lengths of i or fewer of its arcs reduced to zero. Also, let $L_{a,i}$ (to be determined later) be the length of the shortest i-arc path to node a. The algorithm essentially assigns to each node, a, $n+1$ labels, $D_{a,0}, D_{a,1}, \ldots, D_{a,n}$ such that initially it is known that $D_{a,i} \geq L_{a,i}$. The labels are then reduced in such a way as to preserve this initial property until all $D_{a,i} = L_{a,i}$. At this point the $n+1^{\underline{st}}$ label at the sink is the length of the desired path.

Letting S be the source and $\bar{S}$ the sink, the algorithm is:

1. For $i = 0, 1, \ldots, n$, set $D_{S,i} = 0$ and $D_{a,i} = \infty$ for $a \neq S$. Set $k = 0$.

2. Let $l(a, b)$ be the length of arc (a, b). Check each arc (a, b) and:

   a. If $D_{a,k} > D_{b,k} + l(a,b)$ set

   $$D_{a,k} = D_{b,k} + l(a,b)$$

   b. If $D_{b,k} > D_{a,k} + l(a,b)$ set

   $$D_{b,k} = D_{a,k} + l(a,b)$$

   c. If $k \geq 1$ and $D_{a,k} > D_{b,k-1}$ set

   $$D_{a,k} = D_{b,k-1}$$

   d. If $k \geq 1$ and $D_{b,k} > D_{a,k-1}$ set

   $$D_{b,k} = D_{a,k-1}$$

   If none of the above hold, make no changes.

3. Repeat 2 until no more changes can be made. Then if $k < n$, increase k by 1 and go back to 2. If $k = n$, terminate as $D_{\bar{S},n}$ is the length of the desired route.

The desired path itself may then be found by the following procedure:

1. Set $k = 1$, $\bar{S} = a(1)$, and $i(1) = n$.

2. Find a node $a(k+1)$ such that either

   $$D_{a(k+1),i(k+1)} = D_{a(k),i(k)} - l(a(k), a(k+1))$$

   and $i(k+1) = i(k)$ or

   $$D_{a(k+1),i(k+1)} = D_{a(k),i(k)}$$

   and $i(k+1) = i(k) - 1$

3. If $a(k+1) \neq S$, increase k by 1 and go back to 2. Otherwise terminate. $S = a(k+1), \ldots, a(1) = \bar{S}$ is the desired path. The arcs whose lengths are

to be reduced to zero are those arcs, $(a(k), a(k+1))$ such that $i(k) \neq i(k+1)$. The arcs to be removed from the primal network are the arcs which intersect these arcs of the dual.

V. Justification of the Algorithm

The procedure for justifying the algorithm of the last section will be to show first that for all a and all i, $D_{a,i} \geq L_{a,i}$ always and $D_{a,i} = L_{a,i}$ when the algorithm terminates. Then, in the following section, a relationship between the $L_{a,i}$ will be established to help verify the process of tracing out the desired path.

__LEMMA 1:__ $D_{a,i} \geq L_{a,i}$ all a and all i.

__PROOF:__ Assume that at one point of the algorithm $D_{a,i} \geq L_{a,i}$ all a and all i. Suppose that the $i+1^{st}$ label at node a is to be changed to $\bar{D}_{a,i}$. Then there is a node b such that either:

1. $\bar{D}_{a,i} = D_{b,i} + l(a,b)$ or
2. $\bar{D}_{a,i} = D_{b,i-1}$

In the first case any i-arc path to node b of length $L_{b,i}$ plus arc (a,b) is an i-arc path of length less than or equal to $\bar{D}_{a,i}$. In the second case any $(i-1)$-arc path to node b of length $L_{b,i-1}$ plus arc (a,b) with length reduced to zero is an i-arc path to node a of length less than or equal to $\bar{D}_{a,i}$. Thus the relationship $D_{a,i} \geq L_{a,i}$ all a and all i still holds. Initially $D_{S,i} = 0 = L_{S,i}$ and $D_{a,i} = \infty > L_{a,i}$ for $a \neq S$. It follows from induction that $D_{a,i} \geq L_{a,i}$ all a and all i always.

LEMMA 2: After a finite number of iterations, $D_{a,0} = L_{a,0}$ all a and remains at that value for all subsequent iterations.

PROOF: Let $S, a_1, \ldots, a_k, a$ be a shortest route from S to a. After one examination of the arcs,

$$D_{a_1,0} \leq l(S, a_1)$$

After 2 iterations,

$$D_{a_2,0} \leq l(S, a_1) + l(a_1, a_2)$$

After $k+1$ iterations,

$$D_{a,0} \leq l(S, a_1) + \sum_{i=1}^{k-1} l(a_i, a_{i+1}) + l(a_k, a) = L_{a,0}$$

Of course $D_{a,0}$ will remain at $L_{a,0}$ for all subsequent iterations since the $D_{a,0}$ are non-increasing and $D_{a,0} \geq L_{a,0}$ by lemma 1. Thus the lemma holds for any particular node. Let $k(a)$ be the number of iterations required in order that $D_{a,0} = L_{a,0}$. After $\max_a k(a)$ iterations $D_{a,0} = L_{a,0}$ all a.

LEMMA 3: Suppose $S, a_1, \ldots, a_k$ is a shortest i-arc path from S to a_k. If arc (a_{k-1}, a_k) has length zero in this path, then

1. $L_{a_k,i} = L_{a_{k-1},i-1}$

Otherwise

2. $L_{a_k,i} = L_{a_{k-1},i} + l(a_{k-1}, a_k)$

PROOF: Suppose arc (a_{k-1}, a_k) has its length reduced to zero in this path. Then $S, a_1, \ldots, a_{k-1}$ is an $(i-1)$-arc path to a_{k-1} of length $L_{a_k,i}$ and $L_{a_{k-1},i-1} \leq L_{a_k,i}$.

Furthermore, any (i-1)-arc path of length $L_{a_{k-1},i-1}$ and arc (a_{k-1},a_k) with length zero forms an i-arc path to a_k of length $L_{a_{k-1},i-1}$ and the first equality holds. If arc (a_{k-1},a_k) does not have its length reduced to zero, then $S,a_1,\ldots,a_{k-1}$ is an i-arc path to a_{k-1} of length $L_{a_k,i} - l(a_{k-1},a_k)$ and $L_{a_k,i} \geq L_{a_{k-1},i} + l(a_{k-1},a_k)$. Also, any i-arc path to a_{k-1} of length $L_{a_{k-1},i}$ and arc (a_{k-1},a_k) is an i-arc path of length $L_{a_{k-1},i} + l(a_{k-1},a_k)$ and the second equality holds.

LEMMA 4: Suppose $S,a_1,\ldots,a_k$ is the shortest i-arc path from S to a_k. Let (a_m,a_{m+1}) be the last arc in this path to have its length reduced to zero. Then,

$$L_{a_k,i} = L_{a_m,i-1} + \sum_{j=m+1}^{k-1} l(a_j,a_{j+1})$$

PROOF:

$$L_{a_{j+1},i} - L_{a_j,i} = l(a_j,a_{j+1}) \qquad j = m+1,\ldots,k-1$$

$$L_{a_{m+1},i} - L_{a_m,i-1} = 0$$

from lemma 3. Summing these equations gives

$$L_{a_k,i} = L_{a_m,i-1} + \sum_{j=m+1}^{k-1} l(a_j,a_{j+1})$$

THEOREM 5: The algorithm terminates after a finite number of iterations with $D_{a,i} = L_{a,i}$, all a and all i.

PROOF: Suppose that after a finite number of iterations $D_{a,i} = L_{a,i}$ all a and all $i \leq M$. Let $S, a_1, \ldots, a_k$ be an $(M+1)$-arc path of length $L_{a_k, M+1}$ and let (a_m, a_{m+1}) be the last arc in this path to have its length reduced to zero. The after one additional iteration,

$$D_{a_{m+1}, M+1} \leq L_{a_m, M}$$

and after $k-m$ additional iterations,

$$D_{a_k, M+1} \leq L_{a_m, M} + \sum_{j=m+1}^{k-1} l(a_j, a_{j+1}) = L_{a_k, M+1}$$

and the theorem holds for any particular $D_{a, M+1}$. Let $k(a)$ be the number of iterations required in order that $D_{a, M+1} = L_{a, M+1}$. Then after $\max_a k(a)$ iterations $D_{a, M+1} = L_{a, M+1}$ all a. Since $D_{a, 0} = L_{a, 0}$ after a finite number of iterations, it follows from induction that $D_{a,i} = L_{a,i}$ all a and all i after a finite number of iterations and the algorithm terminates.

Of course any route from source to sink with n or fewer of its arcs reduced to zero is an n-arc path to $\bar{S}$. Thus, when the algorithm terminates, $D_{\bar{S}, n} = L_{\bar{S}, n}$ is the length of the desired route. It now remains to justify the procedure for tracing out the desired path. This will be done by showing that the steps of this procedure can be carried out, that the procedure is finite, and finally that the path found is the desired route.

LEMMA 6: Let $L_{a(k), i(k)}$ be the length of the shortest $i(k)$-arc path to node $a(k)$. Then there is a node $a(k+1)$ such that either:

1. $L_{a(k+1), i(k+1)} = L_{a(k), i(k)} - l(a(k+1), a(k))$

where $i(k+1) = i(k)$ or

2. $L_{a(k+1), i(k+1)} = L_{a(k), i(k)}$

where $i(k+.) = i(k) - 1$

(i.e. Step 2 of the procedure for finding the desired route can be carried out.)

PROOF: Let $S, a_1, \ldots, a_m, a(k)$ be an $i(k)$-arc path to node $a(k)$ of length $L_{a(k), i(k)}$. By lemma 3, either

1. $L_{a(k), i(k)} = L_{a_m, (i(k)-1)}$ or
2. $L_{a(k), i(k)} = L_{a_m, i(k)} + l(a_m, a(k))$

Letting $a_m = a(k+1)$ gives the desired relationship.

LEMMA 7: The procedure for tracing the desired path is finite. (i.e. There exists m such that $a(m) = S$.)

PROOF: Either

1. $L_{a(k+1), i(k+1)} = L_{a(k), i(k)} - l(a(k+1), a(k))$

which implies

$$L_{a(k+1), i(k+1)} \leq L_{a(k), i(k)} - \min_{a, b} l(a, b)$$

or

2. $i(k+1) = i(k) - 1$

Suppose the process were not finite. Then there is an M such that $k \geq M$ implies either:

$$L_{a(k), i(k)} < 0 \text{ or } i(k) < 0$$

which is impossible. Thus the process is finite.

THEOREM 8: Let $a(1), \ldots, a(m)$ be the set of nodes found by the tracing procedure. Then $S = a(m), \ldots, a(1) = \bar{S}$

is the shortest possible route if n or fewer arcs have their lengths reduced to zero.

PROOF: Suppose $a(m), \ldots, a(k)$ is an $i(k)$-arc path to $a(k)$ of length $L_{a(k), i(k)}$. Either

1. $L_{a(k-1), i(k-1)} = L_{a(k), i(k)} + l(a(k), a(k-1))$
   where $i(k-1) = i(k)$ or

2. $L_{a(k-1), i(k-1)} = L_{a(k), i(k)}$
   where $i(k-1) = i(k) + 1$.

In either case $a(m), \ldots, a(k-1)$ is an $i(k-1)$-arc path to $a(k-1)$ of length $L_{a(k-1), i(k-1)}$. Since $a(m) = S$ is a path of zero length and hence an $i(m)$-arc path of length $L_{a(m), i(m)}$ to $a(m)$, it follows from induction that $a(m), \ldots, a(1)$ is an n-arc path to $a(1) = \bar{S}$ of length $L_{a(1), i(1)} = L_{\bar{S}, n}$.

This completes the justification of the algorithm for finding the desired route itself. The arcs in this route whose lengths are to be reduced to zero are, of course, those arcs, $(a(k), a(k+1))$ where $i(k) \neq i(k+1)$. The n or less arcs of the original network, which when removed will reduce its capacity the most, are those which intersect these arcs of the dual.

LIST OF SYMBOLS

(a, b)	Arc joining a and b
$l(a, b)$	Length of the arc joining a and b
$D_{a,i}$	$i+1^{st}$ label of node a
$L_{a,i}$	Length of the shortest path from the source to node a if i or fewer arcs have lengths reduced to zero.
$[A, B]$	The cut set which consists of arcs joining nodes in A to nodes in B
$V[A, B]$	Value of cut set $[A, B]$

REFERENCES

1. Ford, L. R., Jr., and D. R. Fulkerson, Flows in Networks, The RAND Corporation, R-375, December 20, 1960.
2. Boldyreff, A. W., Determination of the Maximal Steady-State Flow of Traffic Through a Railroad Network, The RAND Corporation, RM-1532 (ASTIA No. AD 88712), August 5, 1955.
3. Fulkerson, D. R., An Out-of-Kilter Method for Minimal Cost Flow Problems, The RAND Corporation, P-1825, April 5, 1960.
4. Busacker, R. G., and P. J. Gowen, A Procedure for Determining a Family of Minimum Cost Network Flow Patterns, Operations Research Office ORO-TP-15, November, 1960.
5. Brossman, M. W., R. G. Busacker, et al, Computer Assisted Strategic Logistic Planning Transportation Phase, Operations Research Office, ORO-T-393, July, 1961.
6. Dantzig, G. B., and D. R. Fulkerson, Notes on Linear Programming -- Part XXI: On the Max-Flow Min-Cut Theorem of Networks, The RAND Corporation, RM-1418-1 (ASTIA No. 86705), January 1, 1955.
7. Dantzig, G. B., and D. L. Johnson, Maximum Payloads per Unit Time Delivered Through an Air Network, Boeing Scientific Research Laboratories, DI-82-0265, June, 1963.
8. Wollmer, R. D., Some Methods for Determining the Most Vital Link In a Railway Network, The RAND Corporation, RM-3321-ISA, April, 1963.